



UNIVERSIDAD
E

ESCOLA SUPERIOR DE ENXEÑARÍA

Memoria do Traballo de Fin de Grao que presenta

D. Julián Jiménez González

para a obtención do Título de Graduado en Enxeñaría Informática

SuperCars: Desarrollo de videojuego para dispositivos móviles



Xullo, 2015

Traballo de Fin de Grao N°: EI 14/15 - 010

Titor/a: Alma María Gómez Rodríguez

Área de coñecemento: Linguaxes e Sistemas Informáticos

Departamento: Informática

Índice

1. INTRODUCCIÓN.....	2
1.1. DESCRIPCIÓN DEL SISTEMA.....	2
1.2. DESCRIPCIÓN DEL PROCESO DE DESARROLLO	3
2. ANÁLISIS	3
2.1. MODELO DE DOMINIO	3
3. DISEÑO	3
3.1. DIAGRAMAS DE CLASES PARCIALES	4
3.1.1. <i>AppDelegate</i>	4
3.1.2. <i>MainMenu</i>	4
3.1.3. <i>RaceConf</i>	4
3.1.4. <i>Race</i>	5
3.1.5. <i>RaceMenu</i>	5
3.1.6. <i>EndRace</i>	6
3.2. DIAGRAMA DE CLASES TOTAL	7
3.3. DESCRIPCIÓN DE CASOS DE USO	8
3.3.1. <i>Salir del juego</i>	8
3.4. DIAGRAMAS DE SECUENCIA DEL SISTEMA	8
3.4.1. <i>Salir del juego</i>	8
3.4.2. <i>Iniciar partida</i>	9
3.4.3. <i>Iniciar carrera</i>	9
3.4.4. <i>Pausar carrera</i>	9
3.4.5. <i>Reanudar carrera</i>	10
3.4.6. <i>Salir de carrera</i>	10
3.4.7. <i>Mover vehículo</i>	11
3.4.8. <i>Terminar carrera</i>	11
3.4.9. <i>Finalizar partida</i>	12
4. IMPLEMENTACIÓN	12
5. PRUEBAS.....	12

1. Introducción

1.1.Descripción del Sistema

SuperCars es un videojuego de carreras en 2D. En él, el jugador debe mover el vehículo a izquierda y derecha para evitar los obstáculos que van apareciendo en la carretera, así como adelantar a los oponentes.

Se pretende que la aleatoriedad en la generación de obstáculos y oponentes provoque situaciones distintas en cada partida, así como el incremento de dificultad.

1.2. Descripción del proceso de desarrollo

Este proyecto se ha desarrollado siguiendo un modelo iterativo e incremental, ya que la propia naturaleza del proyecto (donde los requisitos y el diseño del sistema no pueden ser detallados con certeza antes de comenzar con su desarrollo), permite que el diseño se vaya adaptando y mejorando a medida que se avanza.

2. Análisis

2.1. Modelo de Dominio

En el Modelo de Dominio de la fase de *Análisis* se reflejan las clases (conceptuales) que se consideran significativas para el dominio del problema. Estas clases no tienen por qué coincidir con las implementadas finalmente, ya que éstas últimas pueden variar en función del entorno de desarrollo e incluso del de despliegue.

La lógica y funcionalidades descritas en estas clases sí deben mantenerse.

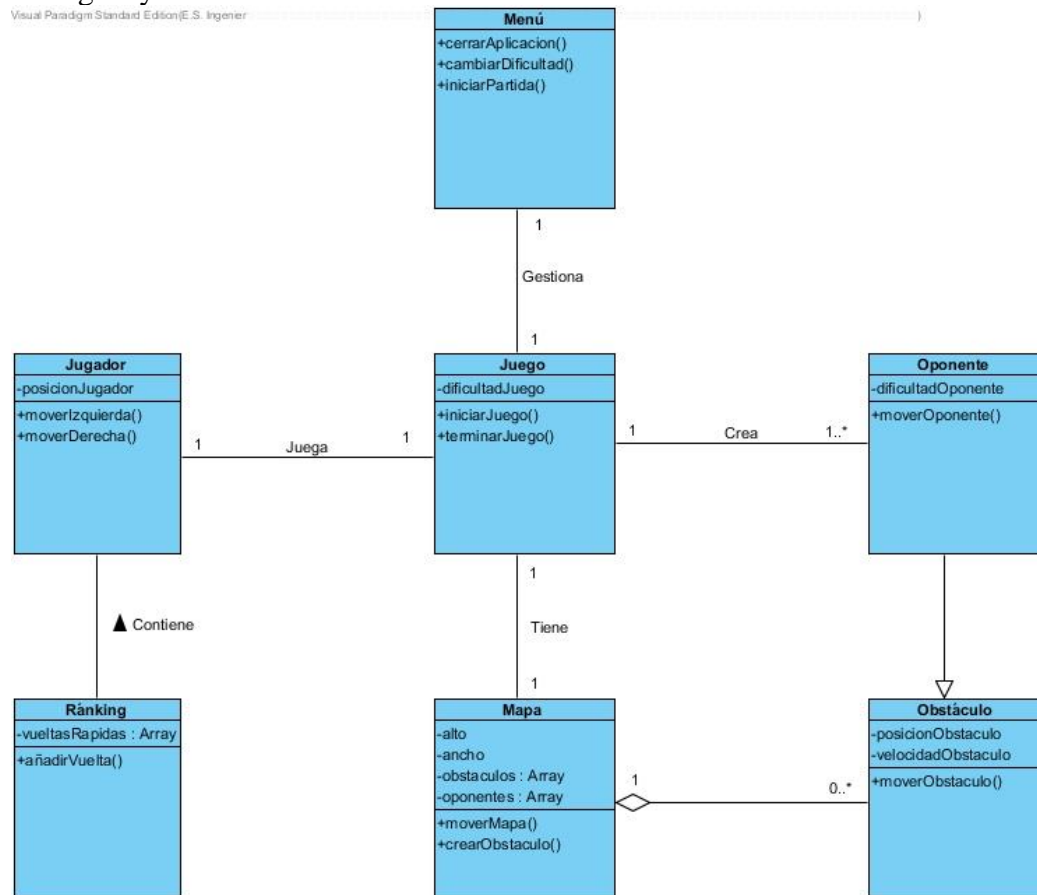


Ilustración 1. Modelo de Dominio de Análisis

3. Diseño

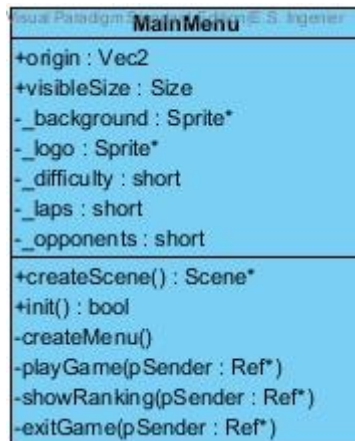
Al iniciar el desarrollo utilizando Cocos2d-x, el diseño final varía ligeramente del diseño de la fase de Análisis (2).

3.1. Diagramas de clases parciales

3.1.1. AppDelegate



3.1.2. MainMenu



3.1.3. RaceConf



3.1.4. Race

Visual Paradigm Standard Edition 3.12.0 - C++	Race
<pre> +origin : Vec2 +visibleSize : Size +audio : CocosDenshion::SimpleAudioEngine* +lapLabel : Label* +posLabel : Label* +playerPos : short +player : cocos2d::Sprite* +leftArrow : cocos2d::Sprite* +rightArrow : cocos2d::Sprite* - _tileMap : TMXTiledMap* - _tileAuxiliarMap : TMXTiledMap* - _speed : float - _difficulty : short - _timeStopped : short - _laps : short - _currentLap : short - _opponents : short - _currentPosition : short - _obstacles : Vector<Sprite*> +createScene() : Scene* +init() : bool +createControls(origin : Vec2, visibleSize : Size) +update(delta : float) +createObstacle(delta : float) +carStopped(delta : float) +moveMap(delta : float) +checkLap(delta : float) +moveObstacles(v : Vector<Sprite*>) +deleteObstacle(s : Sprite*) -onTouchBegan(touch : Touch *, event : Event*) : bool -scheduleAll() -createMenu() -createLapLine() -updateLapsLabel() -updatePosLabel() -checkCollisions(v : Vector<Sprite*>) -showEndRace(pSender : Ref*) -showRaceMenu(pSender : Ref*) </pre>	

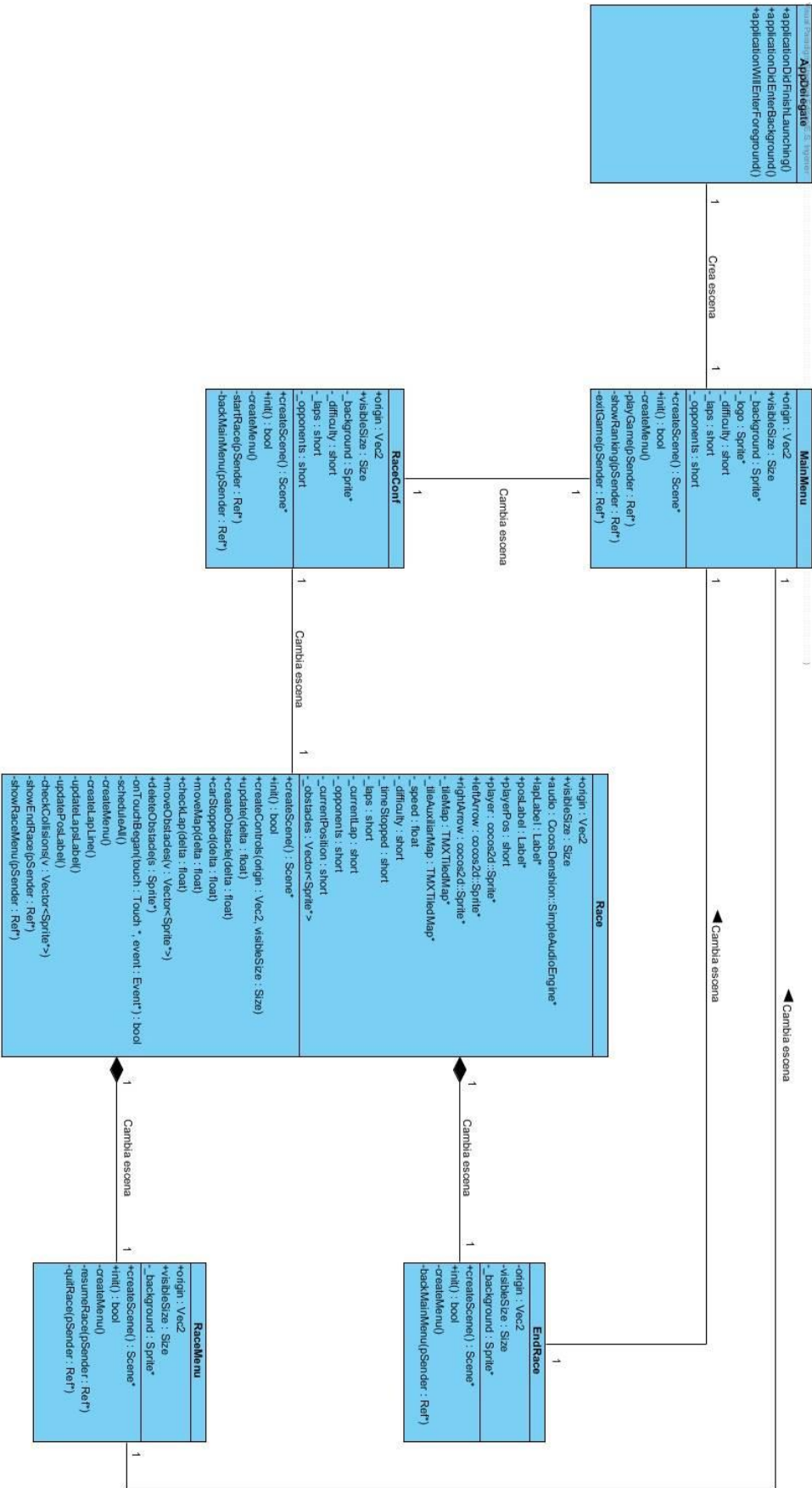
3.1.5. RaceMenu

Visual Paradigm Standard Edition 3.12.0 - C++	RaceMenu
<pre> +origin : Vec2 +visibleSize : Size - _background : Sprite* +createScene() : Scene* +init() : bool -createMenu() -resumeRace(pSender : Ref*) -quitRace(pSender : Ref*) </pre>	

3.1.6. EndRace

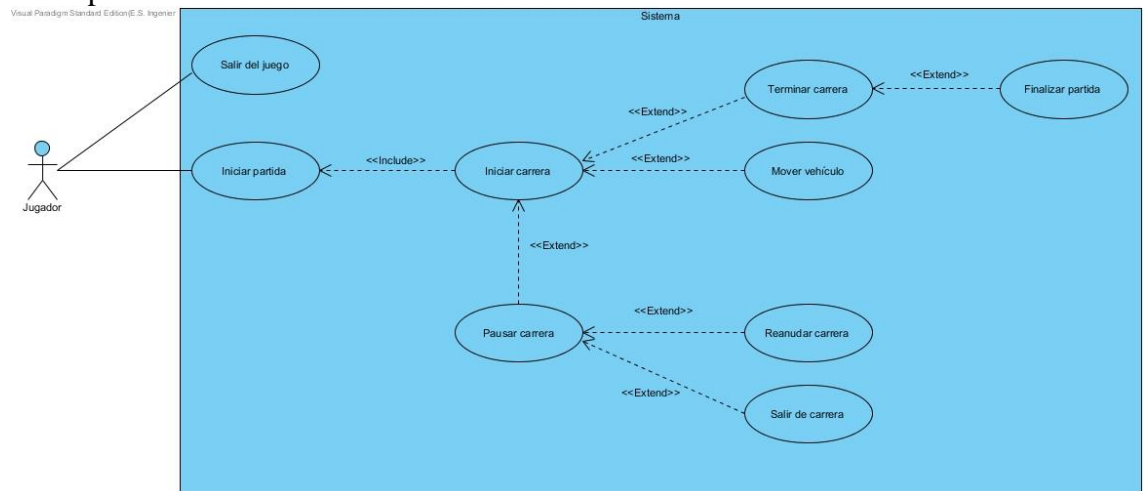
EndRace
<div><div>-origin : Vec2</div><div>-visibleSize : Size</div><div>-_background : Sprite*</div></div>
<div><div>+createScene() : Scene*</div><div>+init() : bool</div><div>-createMenu()</div><div>-backMainMenu(pSender : Ref*)</div></div>

3.2.Diagrama de clases total



3.3.Descripción de Casos de Uso

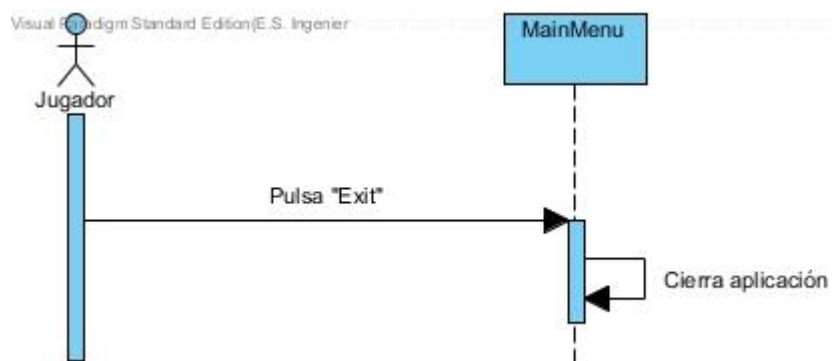
A continuación se muestra el diagrama de casos de uso, seguido de las descripciones detalladas de cada uno de ellos.



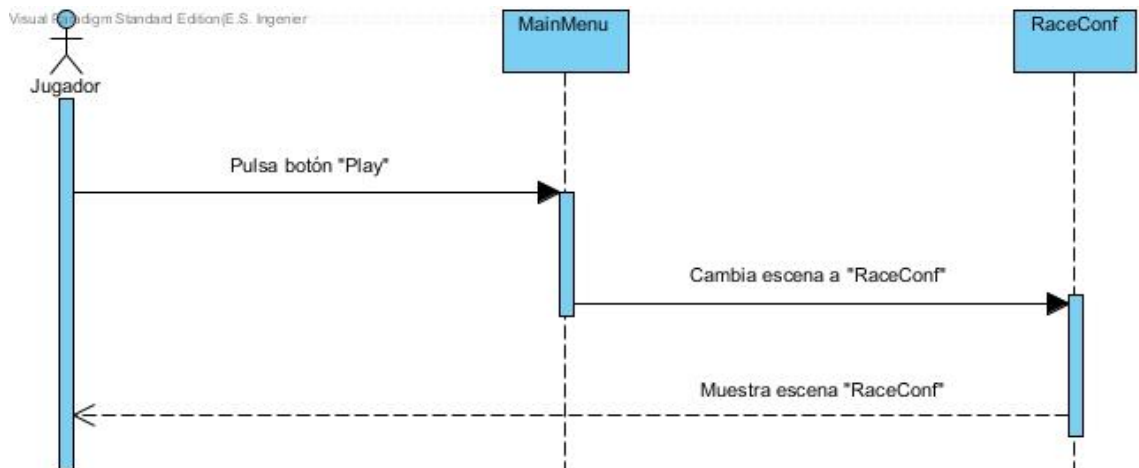
3.3.1. Salir del juego

3.4.Diagramas de Secuencia del Sistema

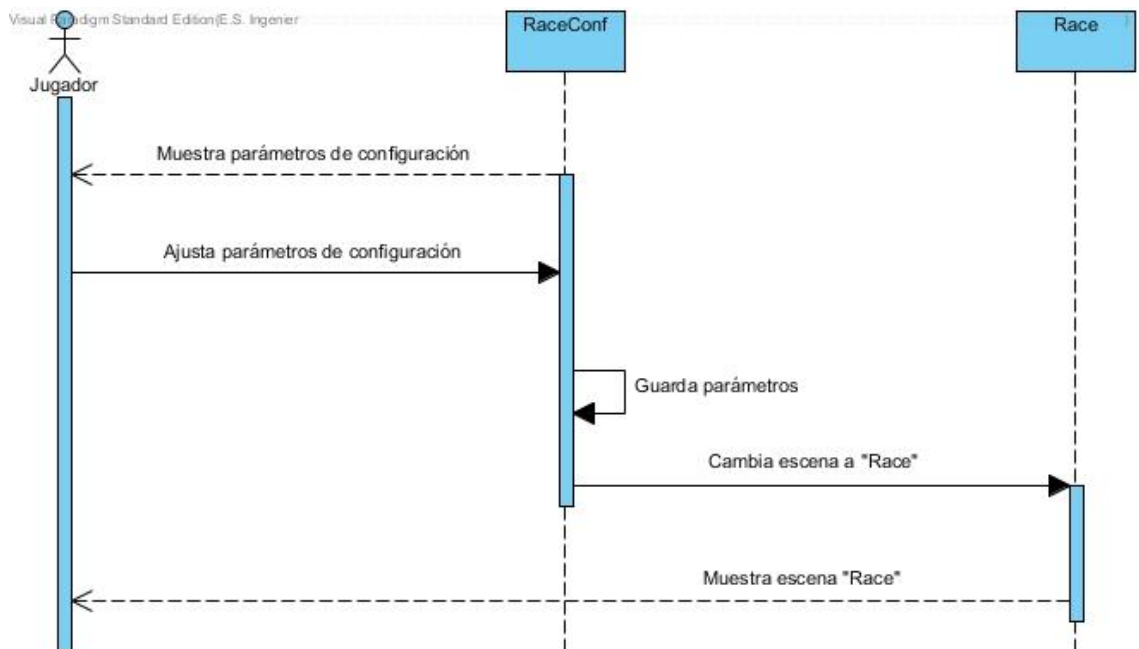
3.4.1. Salir del juego



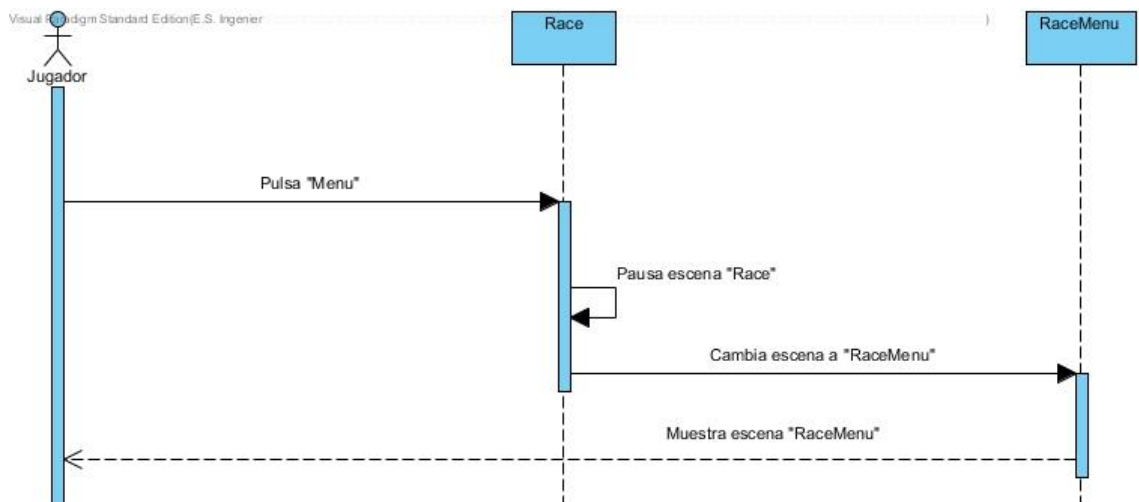
3.4.2. Iniciar partida



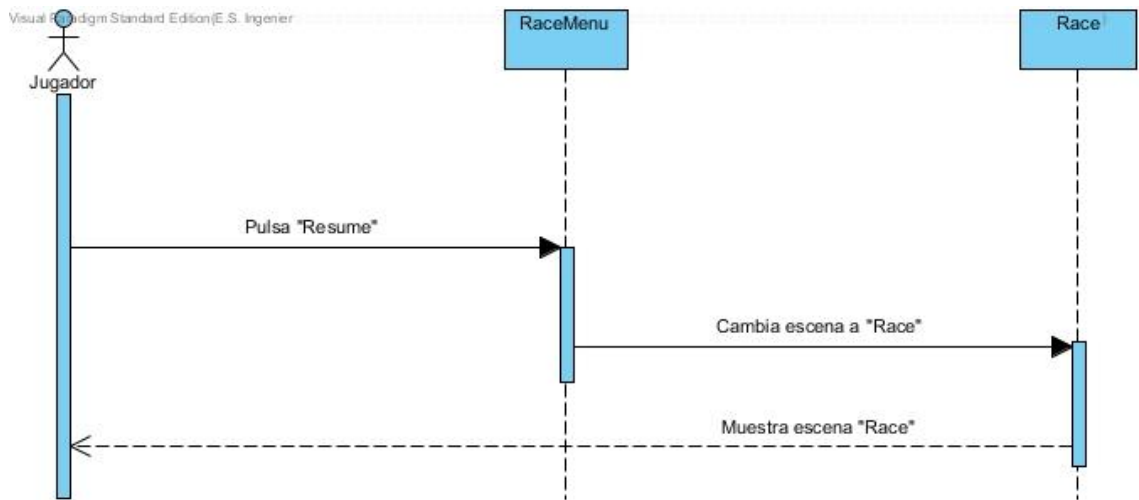
3.4.3. Iniciar carrera



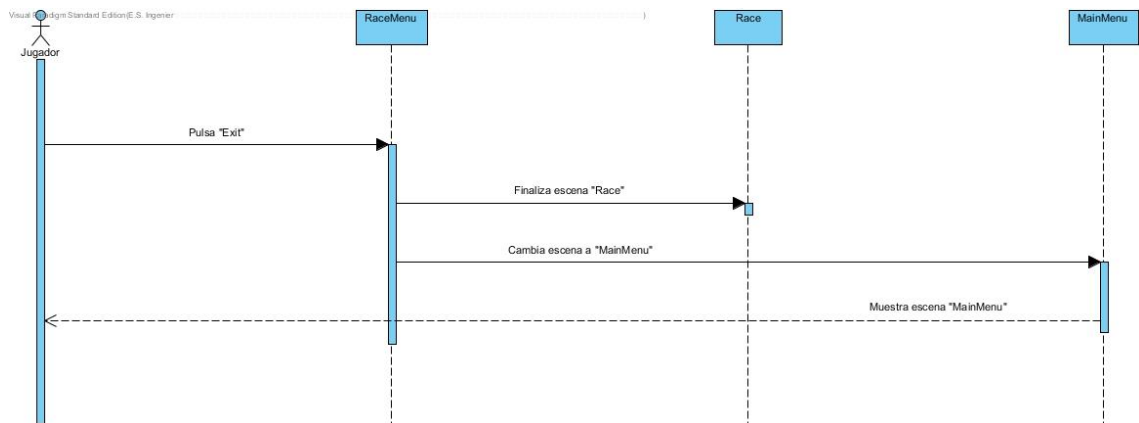
3.4.4. Pausar carrera



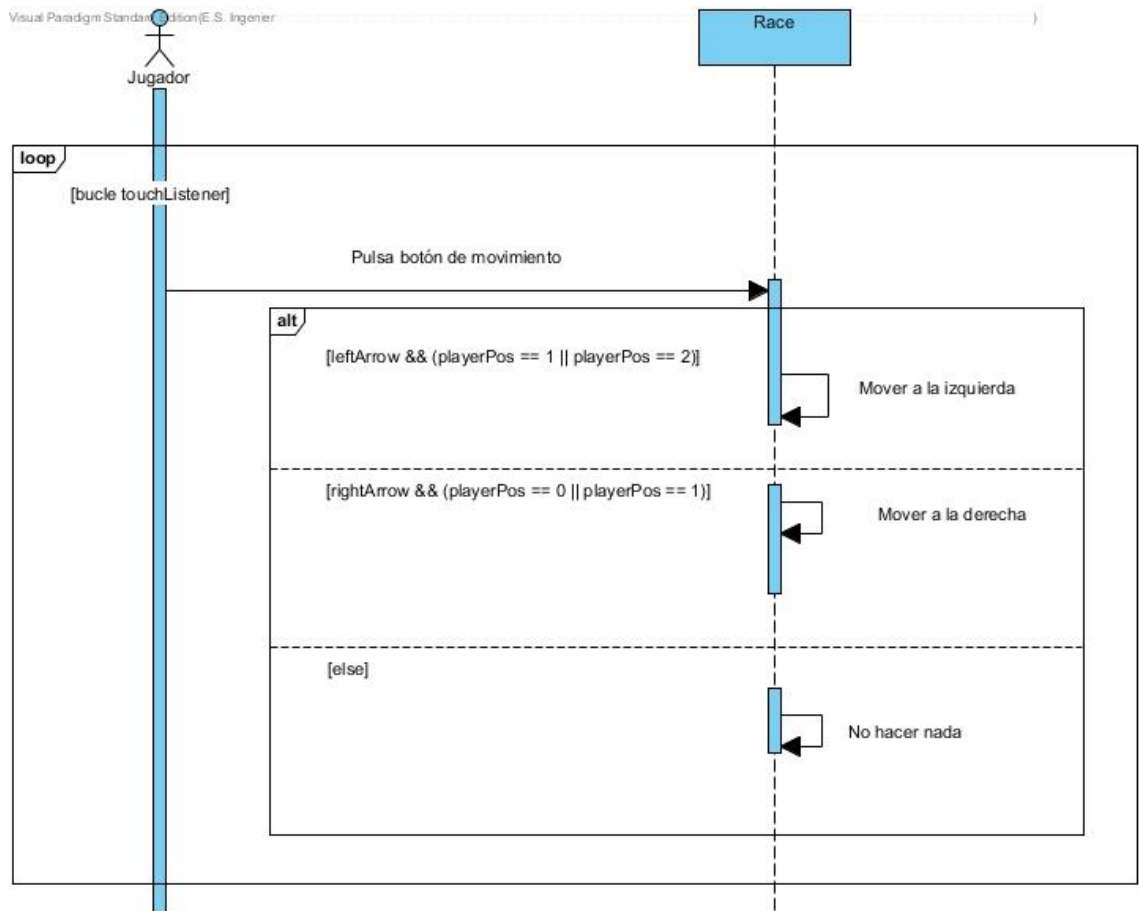
3.4.5. Reanudar carrera



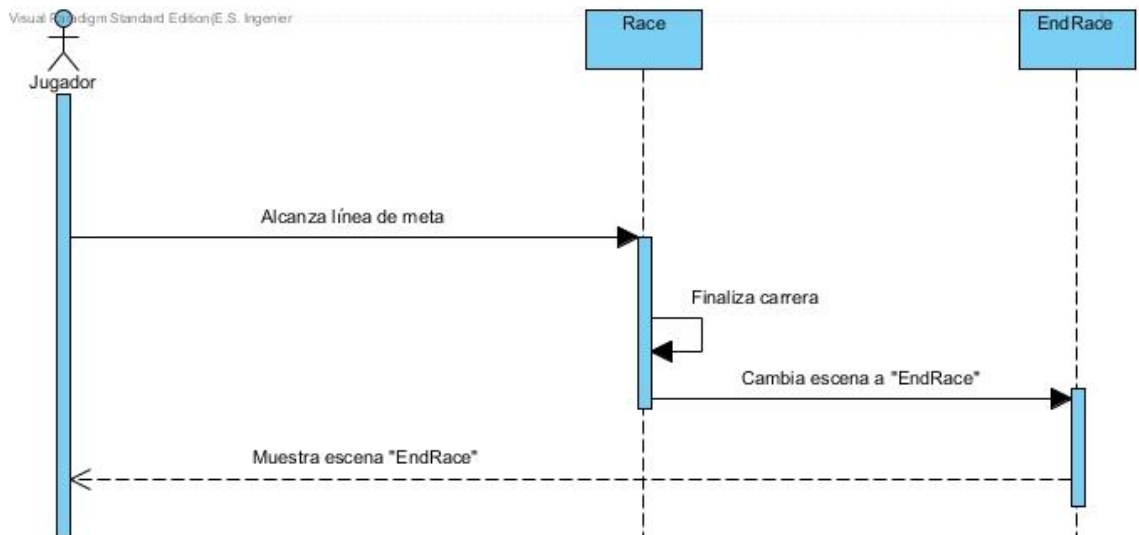
3.4.6. Salir de carrera



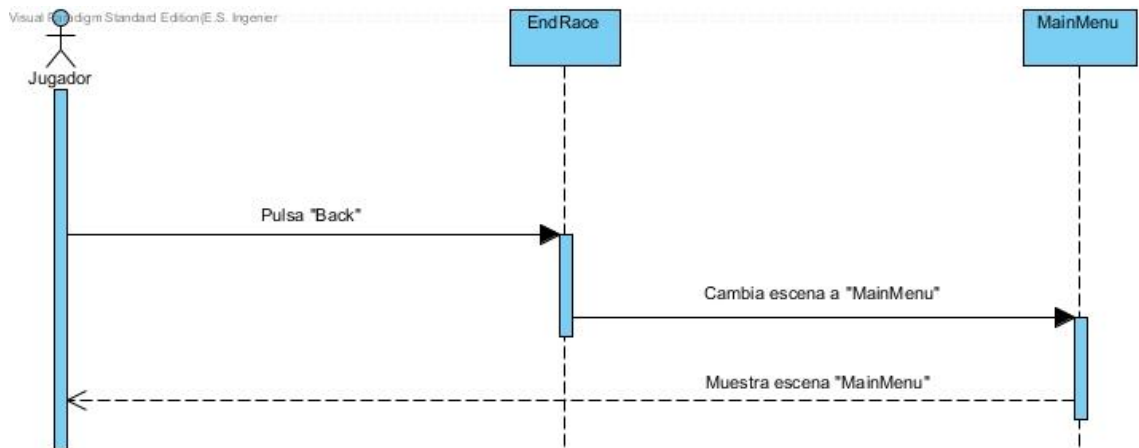
3.4.7. Mover vehículo



3.4.8. Terminar carrera



3.4.9. Finalizar partida



4. Implementación

5. Pruebas
